UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/523,877	03/13/2000	Peter Warnes	ARC.005A	6131

7590 12/16/2002

Gazdzinski & Associates
c/o Robert F Gazdzinski Esq
3914 Murphy Canyon Road
Suite A232
San Diego, CA 92123

EXAMINER
HUISMAN, DAVID J

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 12/16/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/523,877	Applicant(s) WARNES ET AL.
	Examiner David J. Huisman	Art Unit 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 March 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-32* is/are pending in the application.

4a) Of the above claim(s) *6-11,22 and 24* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,12-21,23 and 25-32* is/are rejected.

7)☒ Claim(s) *12 and 17* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 March 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *6 and 8* .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Election/Restrictions

1. Restriction to one of the following inventions is required under 35 U.S.C. 121:

   I. Claims 1-5, 12-21, 23, and 25-32, drawn to a processor and method for executing jump instructions which are configured in any of a plurality of delay slot modes, classified in class 712, subclass 233.

   II. Claims 6-11, 22, and 24, drawn to a method and apparatus for synthesizing a processor design, classified in class 716, subclass 18.

2. Inventions I and II are related as subcombinations disclosed as usable together in a single combination. The subcombinations are distinct from each other if they are shown to be separately usable. In the instant case, invention II has separate utility such as a method for synthesizing the design of a processor. See MPEP § 806.05(d).

3. Because these inventions are distinct for the reasons given above and the search required for Group II is not required for Group I, restriction for examination purposes as indicated is proper.

4. During a telephone conversation with Robert Gazdzinski on December 3, 2002, a provisional election was made without traverse to prosecute the invention of Group I, claims 1-5, 12-21, 23, and 25-32. Affirmation of this election must be made by applicant in replying to this Office action. Claims 6-11, 22, and 24 are withdrawn from further consideration by the examiner, 37 CFR 1.142(b), as being drawn to a non-elected invention.

5. Claims 1-5, 12-21, 23, and 25-32 have been examined.

Papers Submitted

6. It is hereby acknowledged that the following papers have been received and placed of record in the file: #3. Power of Attorney, declaration, and Surcharge as received on 7/12/2000, #4. Request for Correction of Filing Receipt as received on 7/10/2000, #5. Request for Correction of Filing Receipt as received on 8/2/2000, #6. IDS as received on 12/19/2000, #7. Change of Address as received on 4/12/2001, #8. IDS as received on 6/25/2001, and #9. Preliminary Amendment as received on 8/30/2001.

Specification

7. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

8. The disclosure is objected to because of the following informalities: On page 12, Table 3 shows four delay slot modes corresponding to bit patterns 001, 010, 011, and 111. However, on page 11, line 26, the applicant discloses five jump delay slot modes (four plus one reserved). The examiner is not clear how Table 3 shows five jump slot modes and only one reserved.

9. The abstract of the disclosure is objected to because of the following: At the bottom of page 13, please include the serial numbers for the applications that are being incorporated by reference. Correction is required. See MPEP § 608.01(b).

10. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the

following is required: The specification fails to support lines 17-18 on page 26 (claim 20) in that the applicant claims that the at least one data bit (which specifies the jump mode) is used in determining whether to branch or not. However, the specification only discloses that the at least one data bit is used to determine if the delay slot instruction is executed or not. A conditional branch instruction's execution would not depend on the jump mode bits. Instead, only the condition that the instruction depends upon will determine whether the branch is taken or not.

Appropriate correction is required.

Drawings

11. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the following details must be shown or the feature(s) canceled from the claim(s). In lines 17-18 on page 26 (claim 20), the applicant claims that the at least one data bit (which specifies the jump mode) is used in determining whether to branch or not. No new matter should be entered.

12. This application has been filed with informal drawings which are acceptable for examination purposes only. Formal drawings will be required when the application is allowed.

A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

13. Claim 12 is objected to because of the following informalities: On page 24, line 3, please change "synthesize" to --synthesized--. Also, on page 24, line 4, please remove the first occurrence of the word "a".

14. Claim 17 is objected to because of the following informalities: On page 25, line 22, the applicant claims a step of "decoding said at least one branch instruction including one value." Also, on page 25, line 24, the applicant claims a step of "determining whether to execute an instruction within said pipeline in a stage preceding that of said at least one branch instruction based on said one value." However, since at least one data bit is assigned one value (as claimed in line 20, on page 25), it follows that more than one value may be decoded. These steps should be rephrased so that they aren't limited to using just one value.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

15. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

16. Claims 1-5, 12, 14-18, 21, 23, 25, and 29 are rejected under 35 U.S.C. 102(b) as being anticipated by Lee et al., U.S. Patent No. 4,755,966 (herein referred to as Lee).

17. Referring to claim 1, Lee has taught a method of controlling the execution of instructions within a pipelined processor, comprising:

a) providing an instruction set comprising a plurality of instruction words. Lee discloses in column 3, lines 46-47, that each instruction within the instruction set contains a 6-bit opcode, which means a total of 64 instructions could exist within the system. Lee also discloses in column 6, lines 36-39, that the system contains a floating-point unit, which means floating-point instructions would exist.

b) each of said instruction words comprising a plurality of data bits. See column 3, lines 40-42. Each instruction is 32 bits.

c) at least one of said words comprising a jump instruction. Fig.2, component 102, shows the use of a branch (jump) instruction.

d) assigning one of a plurality of values to at least one of said data bits of said at least one jump instruction. See column 3, lines 46-51. Lee discloses that each branch instruction contains a nullify bit and a displacement sign bit. These bits can be assigned a value (0 or 1) as shown in Fig.3.

e) controlling the execution of at least one subsequent instruction within said pipeline based on said one assigned value of said at least one data bit when said at least one jump instruction is decoded. See column 3, lines 58-61. By setting or clearing this nullify bit, a subsequent instruction's execution is controlled.

18. Referring to claim 2, Lee has taught a method as described in claim 1. Lee has further taught that the act of assigning comprises identifying a plurality of data bits within said at least one jump instruction and assigning one of two discrete values to each of said data bits, the combination of said two discrete values representing at least three jump delay slot modes within said processor. See column 5, lines 7-46 for a description of the 5 different jump delay slot

modes shown in Fig.2 In essence, the nullify bit (Fig.1, component 507) and the displacement sign bit (Fig.1, component 508) are checked by the system and the combination of those values will determine the jump delay slot mode. It is inherent that each bit would be assigned one of two discrete values (i.e. 0 or 1) since a processor only understands binary values.

19. Referring to claim 3, Lee has taught a method as described in claim 2. Lee has further taught that the act of controlling the execution based on said discrete values comprises selecting at least one mode from the group comprising:

a) executing said at least one subsequent instruction under all circumstances. See column 5, lines 50-53, and Fig.3.

b) executing said at least one subsequent instruction only if a jump occurs. See column 5, lines 53-57, and Fig.3. The delay slot instruction will be executed when a jump occurs and the displacement is negative. The delay slot instruction will not be executed if a jump does not occur and the displacement is negative.

c) stalling the pipeline or inserting a bubble into the pipeline if a jump occurs. See column 5, lines 32-37, and Fig.2. Note that if a jump occurs and the displacement is positive (as shown in Fig.2, component 112), the delay slot instruction would be fetched but not executed. Therefore, in order to kill the unwanted instruction, the pipeline would be stalled for at least a single cycle.

20. Referring to claim 4, Lee has taught a method as described in claim 3. Lee has further taught that at least one jump instruction comprises a conditional branch instruction. See column 2, lines 62-64, and note the conditional component 202.

21. Referring to claim 5, Lee has taught a method as described in claim 1. Furthermore, note that the displacement sign bit is assigned one value as is the nullify bit. Therefore, claim 5 is rejected for the same reasons set forth in the rejection of claim 3 above.

22. Referring to claim 12, it has been noted by the examiner that the only difference between claim 12 and claim 3 is that claim 3 claims a method whereas claim 12 claims a machine-readable data storage device that stores a program which performs the method of claim 3. Therefore, the instruction set (with multiple jump modes) of claim 12 is rejected for the same reasons set forth in the rejection of claim 3. Furthermore, it is inherent that a processor executes programs which comprise a plurality of bits. It is further inherent that the program must be stored in some data storage medium. For instance, for a processor to execute a program, the instructions must reside in the processor's memory. Also, the processor may have access to a hard-disk data storage device where the program may be stored. If this were the case, the program would be transferred from the hard-disk to the memory before execution.

23. Referring to claim 14, Lee has taught a digital processor comprising:

a) a processor core having a multistage instruction pipeline, said core being adapted to decode and execute an instruction set comprising a plurality of instruction words. Fig.5 shows a 4-stage pipeline that is further described in column 6, line 56, to column 7, line 39. Furthermore, it is inherent that the processor will decode and execute multiple instructions (as established in the rejection of claim 1) from an instruction set.

b) a data interface between said processor core and an information storage device. It is inherent that in order for a processor to execute instructions, they must be stored in the processor's

memory. Therefore, the instructions would be stored in an information storage device that is directly accessible by the processor.

c) an instruction set comprising a plurality of instruction words, at least one of said instruction words being a jump instruction containing data defining a plurality of jump delay slot modes, said plurality of modes controlling the execution of instructions within said instruction pipeline of said processor core in response to said at least one jump instruction word within said instruction set. This portion of claim 14 is rejected for the same reasons set forth in the rejection of claim 3 above.

24. Referring to claim 15, Lee has taught a method as described in claim 14. Furthermore, claim 15 is rejected for the same reasons set forth in the rejection of claim 3 above.

25. Referring to claim 16, Lee has taught a method as described in claim 14. Lee has further taught that at least one jump instruction comprises a conditional branch instruction having an associated logical condition, the execution of a jump to the address within said information storage device specified by said at least one conditional branch instruction being determined by said logical condition. See Fig.3, component 202, and note that the branch will be taken or not taken based on some condition. Furthermore, it is inherent that the execution of the branch will cause a jump to the address (specified by the instruction) within the information storage device.

26. Referring to claim 17, Lee has taught a digital processor having at least one pipeline and an associated data storage device, wherein the execution of instructions within said at least one pipeline is controlled by the method comprising:

a) storing an instruction set within said data storage device, said instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits,

at least one of said instruction words comprising a branch instruction directing branching to a first address within said data storage device. As discussed above in the rejections of claim 12 and 14, Lee has taught an instruction set with a plurality of instructions that would inherently comprise a plurality of bits and would inherently be stored in a data storage device in order to processor-accessible and executable. Furthermore, from the rejection of claim 16, it is inherent that a branch will cause a jump to a specified address within the data storage device.

b) assigning one of a plurality of values to at least one of said data bits of said at least one branch instruction. See column 3, lines 46-51. Lee discloses that each branch instruction contains a nullify bit and a displacement sign bit. These bits can be assigned a value (0 or 1) as shown in Fig.3.

c) decoding said at least one branch instruction including said one value. Since, the values of the nullify bit and displacement sign bit are part of each branch instruction, it follows that the value will be decoded as the branch instruction is decoded. See Fig.1 and column 3, lines 46-51.

d) determining whether to execute an instruction within said pipeline in a stage preceding that of said at least one branch instruction based on said one value. The delay slot instruction would be in a pipeline stage preceding that of the branch instruction. And, Lee discloses a system in which the execution of the delay slot instruction is determined based on the value of the nullify bit and/or displacement bit.

e) branching to said first address based on said at least one branching instruction. Recall from claim 16, that it is the inherent nature of a branch instruction (when taken) to jump to a specified address.

27. Referring to claim 18, Lee has taught a processor as described in claim 17. Furthermore, it is inherent that the data bits comprise binary data. A processor can only recognize and understand zeroes and ones.

28. Referring to claim 21, Lee has taught a method as described in claim 20. Furthermore, claim 20 is rejected for the same reasons set forth in the rejection of claim 3 above.

29. Referring to claim 23, Lee has taught a digital processor comprising:

a) processing means having a multistage data pipeline, said processing means being adapted to decode and execute an instruction set comprising a plurality of instruction words. Recall that Lee has disclosed a multi-stage pipeline in Fig.5 and column 6, lines 59-66. Also the decoding and executing is disclosed in column 7, lines 12-19. Finally, it has been established in the rejection of claim 1 above that Lee has taught an instruction set with a plurality of instruction words.

b) means for storing data. It is inherent that the processor must have a memory from which instructions can be read. Also, Lee has taught that the system includes a register file for temporary storage. See column 6, lines 36-40.

c) data interface means for transferring data between said processing means and said means for storing data. In order for instructions to be processed, the processor must fetch them from memory first. Therefore, it is inherent that an interface exists between the processor and the memory.

d) an instruction set comprising a plurality of instruction words, at least one of said instruction words being a jump instruction containing data defining a plurality of jump control means, said plurality of jump control means controlling the execution of instructions within said data pipeline

of said processing means in response to said at least one jump instruction word within said instruction set. This portion of claim 23 is rejected for the same reasons set forth in the rejection of claim 14 above.

30. Referring to claims 25 and 29, Lee has taught a method and digital processor as described in claims 1 and 14, respectively. Lee has further taught that at least one of said plurality of instruction words comprises an op-code and a plurality of fields, each of said fields comprising a plurality of bits (see column 3, lines 46-51), said at least one instruction word being encoded according to the method comprising:

a) associating a first of said fields with a first data source. See column 3, lines 47-48 (component 503).

b) associating a second of said fields with a second data source. See column 3, lines 48-49 (component 504).

c) performing a logical operation using said first and second data sources as operands, said logical operation being specified by said op-code. See column 3, lines 51-55. In this case, the opcode specifies a compare and branch instruction where the comparison is performed between the contents of the two specified registers.

Claim Rejections - 35 USC § 103

31. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

32. Claims 13 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee, as applied to claim 12 above.

33. Referring to claim 13, Lee has taught a storage device as described in claim 12. Lee has not taught that the storage medium is a compact disk-read only memory (CD-ROM). Official notice is taken that the concept of using CD-ROMs as a type of data storage medium is well known and expected in the art. It would have been obvious to use a CD-ROM to store a program in order to reduce the consumption of internal hard-disk resources of the computer. In addition, using a CD-ROM to store a program allows for mobility of the storage medium, meaning the program can be executed on multiple, independent machines by loading the CD-ROM into each machine.

34. Referring to claim 19, Lee has taught a processor as described in claim 17. Lee has further taught a 4-stage pipeline with an instruction address generation stage, an instruction fetch stage, an execute stage, and a write stage. See Fig.5 and column 6, lines 56-64. Lee has not explicitly taught a stage just for decoding. However, Lee has taught that decoding is done in one of the aforementioned stages. See column 7, lines 12-16. Lee further states that the execution of instructions can be pipelined to any depth desired. See column 6, lines 63-64. It is well known in the art that pipelines include a separate decode stage. The actual implementation of the pipeline is a designer's preference but Lee has taught a system in which any size pipeline would suffice. Therefore, if desired, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a pipeline with a separate decode stage, as is well known in the art.

35. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lee in view of Nguyen et al., U.S. Patent No. 5,838,984 (herein referred to as Nguyen).

36. Referring to claim 20, Lee has taught a method of controlling the branching within the program of a multi-stage pipelined digital processor, comprising:

a) storing an instruction set within said data storage device, said instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said instruction words comprising a branch instruction directing branching to a first address within said data storage device based on a first parameter. This portion of claim 20 is rejected for the same reasons set forth in the rejection of claim 17 above. Furthermore, the first parameter could be considered the logic that determines whether the branch is taken or not taken (ex. a flag is checked and compared to a value).

b) Lee has not explicitly taught the defining of jump modes that are assigned to at least one data bit of the branch instruction, wherein the at least one data bit is decoded and used in the determination of the branch outcome along with a first parameter. However, Nguyen has taught such a concept. See Fig.8 and column 11, lines 58-67. Note that Nguyen has disclosed multiple jump modes (i.e. branch if equal to, branch if greater than, branch if less than, branch always, etc.) and these jump modes are denoted by setting the appropriate bits within the condition field of the branch instruction. See Fig.8 (CT format). As previously discussed, since the condition bits are part of the branch instruction, they will be decoded along with the branch instruction and these bits in combination with a first parameter (i.e. a flag as mentioned in part (a)) would be used in determining whether or not the branch will be taken or not taken. A person of ordinary skill in the art would have recognized that by having this condition field, multiple branch types

would be possible while keeping the opcode constant, resulting in more flexibility for the programmer. Also, the condition field can also be used for other instructions as shown in the bottom two formats of Fig.8 and column 12, lines 1-16. For instance, a conditional move can be implemented in which a move would only occur if a certain condition were met. Again, this would give the programmer more programming options in that more functionality is available with such instructions (since multiple conditions could be checked). Therefore, in order to increase the programmer's flexibility, it would have been obvious to one of ordinary skill in the art at the time of the invention to assign multiple jump modes to at least one data bit in the branch instruction, wherein the at least one data bit is decoded and used in the determination of the branch outcome along with a first parameter.

37. Claims 26-28 and 30-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee, as applied to claims 1 and 14, respectively, in view of Kawasaki et al., U.S. Patent No. 5,530,965 (herein referred to as Kawasaki).

38. Referring to claims 26, 27, 30, and 31, Lee has taught a method and digital processor as described in claims 1, 25, 14, and 29, respectively.

a) Lee has further taught of providing an instruction word having an opcode and at least one short immediate value associated therewith, said at least one short immediate value comprising a plurality of bits. Note from column 3, lines 46-51, that the instruction format includes an 11 bit displacement field, which holds an immediate constant for branching purposes.

b) Lee has not explicitly taught selecting a portion of said plurality of bits of said at least one short immediate value, shifting all of said bits of said at least one short immediate value using

said opcode and only said portion of bits to produce a shifted immediate value, and storing said shifted immediate value in a register. However, Kawasaki has taught such a concept for branching purposes as well as for other instructions, such as move instructions. See column 51, lines 50-55. Kawasaki has disclosed a move instruction of the format: **mov #imm, Rn**, where the short immediate value specified by #imm is sign-extended to form a long immediate value which is then stored in the register specified by Rn. By sign-extending an immediate value, a portion (the sign bit) of the value is copied into the most significant bit positions of the long immediate value. For instance, if the #imm field specified a short 4-bit immediate value 1010, which is to be transformed into an 8-bit long immediate value, then 1010 would be sign-extended to 11111010, where the sign bit of the 4-bit value is copied into the 4 most significant bit positions of the long value. It also follows that the 4-bit value has been shifted. Note that initially, 1010 contained a 1 in the most significant bit position, a 0 in the second most significant bit position, a 1 in the third most significant bit position, and a 0 in the fourth most significant bit position. After sign-extending the 4-bit value, the same numbers become the fifth, sixth, seventh, and eighth most significant bits, respectively. Hence, they have been shifted. Furthermore, in column 42, lines 16-27, Kawasaki has disclosed that this type of move instruction is used to help branch to addresses out of the short immediate value's range. More specifically, if a branch needs to branch further than what the short displacement allows, then the destination address is moved to the register specified by the "mov" instruction and a "jmp" instruction (shown in column 48, lines 28-30) is used with to jump to the address stored in the register. A person of ordinary skill in the art would have recognized that this concept could be applicable in a system that is concerned with branching, such as Lee's. Such a concept would

allow a branch instruction to branch to an address outside of the range of just a short immediate displacement value. This in turn would give a programmer more freedom in that they would not have to worry about program length or certain parts of a program being out of reach of a branch. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to take Kawasaki's concept of selecting a portion of said plurality of bits of said at least one short immediate value, shifting all of said bits of said at least one short immediate value using said opcode and only said portion of bits to produce a shifted immediate value, and storing said shifted immediate value in a register, and apply it to the system of Lee.

39. Referring to claims 28 and 32, Lee in view of Kawasaki has taught a method and digital processor as described in claims 27 and 31, respectively. Recall that Lee has taught an instruction format that includes a 6-bit opcode field, meaning Lee's system has the ability of choosing between 64 different instructions if necessary. With the exception of branching, Lee has not explicitly stated any other instructions that have been implemented. However, it is inherent that other instructions would exist so that the processor can perform useful operations. A processor that does nothing but branching would do nothing useful. Furthermore, in column 3, lines 46-51, Lee has disclosed that his system is a register-register (load-store) architecture, i.e. where main memory is only accessed through load and store operations and operations are performed on values in registers. This is known because Lee has implemented an instruction format with two register operands. A move (mov) instruction is also well known in the art and is explicitly shown in Kawasaki. More specifically, Kawasaki has shown multiple versions of a move instruction. One version moves a short immediate value into a register (as shown in column 51, lines 51 and 55-56) and another version includes moving a value from one register to

another (as shown in column 52, line 53). A person of ordinary skill in the art would expect to find these common move instructions within the system of Lee because they allow a programmer to move an initial value into a register as well as temporarily store a register value into another register. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to have at least one instruction word having a plurality of fields and said at least one instruction word having a short immediate value comprise the same instruction word(s); in this case, the move instruction, taught by Kawasaki.

Conclusion

40. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

Hennessy & Patterson, Computer Architecture A Quantitative Approach, 2nd Edition, Morgan Kaufmann Publishers, Inc. 1996, pages 70-71 & 132, has taught the background of the register-register architecture as well as a common 5-stage pipeline including a decode stage.

Dulong et al., U.S. Patent No. 4,974,155, has taught a variable delay branch system in which branch delay slots are filled in order to prevent breaks in a pipeline.

Leung et al., U.S. Patent No. 5,784,603, has taught the fast handling of branch delay slots on mispredicted branches. Leung et al. has also taught that the Sun Sparc processor includes a

annul bit within the branch instruction which is used to prevent the delay slot instruction from executing if necessary.

Cherabuddi et al., U.S. Patent No. 6,256,729,has taught a method and apparatus for resolving multiple branches. An annul bit is also disclosed which is used to invalidate a delay slot instruction.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

DJH
David J. Huisman
December 9, 2002

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100